No announcements today! (Hope quiz 2 went well!)

Last time:

- Review of Church-Turing thesis and the move to high-level description.
- Review of how to show L is TM-recognizable or TM-decidable.
- TM-decidable closed under complement (proved previously), but TM-recognizable is not (will prove later)
- Thm: The class of TM-decidable languages is closed under complement
- Thm: The class of TM-recognizable languages is closed under union.

} exercise: what about intersection?

- We used this ↑ to introduce the idea of running in parallel and why it is different than sequential, for general algorithms (that may not always halt)
- Theorem: L is decidable $\iff$ both L and $\overline{L}$ are recognizable
- Discussed encodings: we denote $\langle B \rangle$ for the encoding of object B (eg, an algorithm like DFA/NFA/TM, or a number, graph, ...) Importantly, these objects ↑ can be encoded in a way that's easy to check/decode (so can be fed as inputs to a TM/algorithm).
- Defined $A_{DFA}$, $A_{NFA}$, $E_{DFA}$ — we'll see these are decidable.

Claim: $A_{DFA} = \{ \langle D, w \rangle \mid D \text{ is a DFA, } w \text{ is a string, and } D \text{ accepts } w \}$ is decidable.

"Acceptance problem for DFAs"

Proof:

Construct a decider algorithm for $A_{DFA}$:

$M_{A_{DFA}}$: "On input $x$,

1. check that $x = \langle D, w \rangle$ is a valid encoding of a DFA $D$, and string $w$
   if not - reject.
2. Run $D$ on $w$
   if $D$ accepts $w$ - accept
   if $D$ rejects $w$ - reject "

Analysis: exercise (fill in ...) (William's lecture 13 notes have solution)

$x \in A_{DFA} \Rightarrow$ $x = \langle D, w \rangle$ where $D$ is a DFA, $w$ string, and $D$ accepts $w$ $\Rightarrow \ldots \Rightarrow$ $M_{A_{DFA}}$ accepts $x$ "

$x \notin A_{DFA} \Rightarrow \ldots \Rightarrow$ $M_{A_{DFA}}$ rejects $x$

Convention: Step 1 above can be abbreviated as:

"On input $\langle D, w \rangle$ where $D$ is a DFA and $w$ is a string, do:"

:

Claim: $A_{NFA} = \{\langle D, w \rangle \mid D \text{ is an NFA, } w \text{ is a string, and } D \text{ accepts } w\}$ is decidable.

"Acceptance problem for NFAs"

proof:

Here's a decider for $A_{NFA}$:

"On input $\langle N, w \rangle$ where $N$ is an NFA and $w$ is a string:

(by convention above: implicit is that if input is wrongly formatted, reject)

$M_{A_{NFA}}$:

1. Convert $\langle N \rangle$ to an equivalent DFA $\langle D \rangle$ using subset construction.
2. Run $M_{A_{DFA}}(\langle D, w \rangle)$
   if it accepts, accept
   if it rejects, reject"

Every step is implementable (we saw alg for (1) previously in class, and for (2) above).

and this decides $A_{NFA}$

Analysis: exercise.

claim: $E_{DFA} = \{\langle D \rangle \mid D \text{ is a DFA, and } L(D) = \emptyset\}$ is decidable.

"Emptiness problem for DFAs"

(we'll see right proof next time)

Discussion / wrong proof for above:

On input DFA $\langle D \rangle$, we want to decide whether it's true that D rejects all inputs.

How about this algorithm:

M: "on input $\langle D \rangle$,
Try running D on all possible inputs
if any is accepted, reject"

Q: Is M a valid algorithm?

A: need to write the first step more clearly, so it's clear it's implementable, but yes, there's a way to algorithmically go through all possible inputs (eg. in lexicographic order)

Q: Is M a decider?

A: NO. It can run forever as we have an infinite loop (go through every input), and we are not guaranteed to eventually reach an input where the algorithm halts.

Q: Is M a recognizer for the $E_{DFA}$? (i.e., what is $L(M)$?)

A: exercise!

For next time, think about decider for $E_{DFA}$, as well as

$$EQ_{DFA} = \{ \langle D_1, D_2 \rangle \mid D_1, D_2 \text{ are DFAs, and } L(D_1) = L(D_2) \}$$

Equivalence problem for DFAs.

What about the corresponding acceptance/emptiness/equivalence problems for TMs? i.e.

For each of $A_{TM}$, $E_{TM}$, $EQ_{TM}$:

Can you show a decider?

Can you show a recognizer?